

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berhubungan dengan teori-teori yang digunakan dalam skripsi ini. Kajian pustaka membahas mengenai penelitian-penelitian yang sebelumnya telah ada dan berkaitan dengan penyusunan skripsi ini. Sedangkan untuk dasar teori selanjutnya disusun berdasarkan latar belakang dan rumusan yang terkait dengan teori apapun untuk mendasari penelitian ini diantaranya adalah penyakit *Chronic Kidney Disease* (CKD), Jaringan Saraf Tiruan (JST), *Extreme Learning Machine* (ELM) dan *Confusion Matrix*.

2.1 Kajian Pustaka

Kajian Pustaka pada penelitian ini akan membahas tentang beberapa penelitian yang sebelumnya telah dilakukan. Beberapa penelitian tersebut akan digunakan peneliti untuk mendukung penelitian ini. Referensi pertama adalah penelitian yang dilakukan oleh Humaini (2015) tentang penerapan metode *Extreme Learning Machine* (ELM) untuk memprediksi kondisi cuaca di wilayah malang. Penelitian tersebut bertujuan untuk mendapatkan arsitektur yang optimal dengan memperhatikan beberapa atribut seperti tekanan udara, suhu, angin dan kelembapan udara yang dapat memprediksi cuaca di wilayah malang. Di dapatkan kesimpulan pada penelitian ini bahwa metode *Extreme Learning Machine* (ELM) dapat menghasilkan nilai akurasi *testing* sebesar 80% memenuhi kriteria data yang ada.

Penelitian kedua dilakukan oleh Pangaribuan (2016) yang menerapkan metode *Extreme learning Machine* (ELM) terhadap penyakit diabetes mellitus. Penelitian tersebut bertujuan untuk membuat sistem peramalan baru yang dapat menjadi alat bantu dalam penentuan apakah seseorang menderita diabetes mellitus atau tidak. Hasil dari penelitian membandingkan antara metode *Extreme Learning Machine* (ELM) dengan metode *backpropagation* menghasilkan nilai MSE pada ELM sebesar 0,4036 dan nilai MSE pada *backpropagation* sebesar 0,9425.

Penelitian ketiga dilakukan oleh Sari (2017) yang menerapkan metode *Extreme Learning Machine* (ELM) untuk peramalan data *time series*. Penelitian tersebut bertujuan untuk memprediksi harga penutupan harian saham BRI yang diperdagangkan pada bursa efek Jakarta. Hasil dari penelitian terlihat bahwa metode ELM memiliki rata rata waktu belajar lebih cepat, dengan kata lain ELM lebih akurat dan cepat dalam proses pembelajaran dengan nilai 0,38309 detik dan nilai RMSE sebesar 21,58585.

Penelitian keempat dilakukan oleh Rahma, Wijaya, dan Prawito (2016) menggunakan metode ELM sebagai alat bantu klasifikasi Stroke Iskemik Akut dan Normal dan di dapatkan bahwa hasil akurasi pelatihan ELM berdasarkan nilai fitur BSI adalah diatas 98% dan hasil akurasi pengujian ELM diatas 85%. Dengan ini metode ELM memiliki akurasi yang tinggi dan waktu pelatihan yang lebih cepat.

Penelitian kelima dilakukan oleh Rianto dan Satvika (2017) menggunakan algoritme C4.5 dalam mendeteksi Penyakit Ginjal Kronis dan di dapatkan hasil

proses pengujian yang dilakukan dengan menggunakan *cross-validation* dan berdasarkan hasil yang telah dihitung aplikasi ini memiliki akurasi 91,50% pada saat *decision tree* dibuat tanpa menggunakan *preprocess* menu.

Penelitian keenam dilakukan oleh Santosa, Widjanarko, dan Supriyanto (2016) menggunakan RBF dalam memodelkan prediksi penyakit ginjal kronik. Didapatkan hasil pengujian yang memiliki akurasi 93,75% pada parameter Learning Rate 0,2 dan Epoch sebesar 2000 lebih baik dibandingkan dengan penelitian sebelumnya dengan menggunakan metode BP yang menghasilkan akurasi sebesar 91,71%.

Penelitian yang terakhir dilakukan oleh Huang, Zhu, dan Siew (2006) menjelaskan tentang metode *Extreme Learning Machine* (ELM) dan terdapat hasil perbandingan metode ELM dengan beberapa metode lainnya seperti SVM, SAOCIF, Cascade-Correlation, AdaBoost, C4.5, RBF dan Heterogeneous RBF. Penerapan perbandingannya dilakukan dalam aplikasi diagnosis penyakit diabetes. Didapatkan hasil metode ELM mendapatkan nilai *Testing Rate* sebesar 77,57% yang dimana metode ELM ini memiliki *Testing Rate* yang paling besar dibandingkan dengan metode-metode lainnya.

2.2 Chronic Kidney Disease (CKD)

Chronic Kidney Disease (CKD) adalah suatu keadaan menurunnya fungsi ginjal yang bersifat *kronik, progresif* dan menetap berlangsung. Beberapa tahun pada keadaan ini ginjal kehilangan kemampuannya untuk mempertahankan volume dan cairan tubuh dalam keadaan asupan diet normal. *Glomerulonephritis* dalam beberapa bentuknya merupakan penyebab paling banyak yang mengawali penyakit ini. Kemungkinan disebabkan oleh terapi *glomerulonephritis* yang agresif dan disebabkan oleh perubahan praktek program penyakit ginjal tahap akhir yang diterima pasien, diabetes mellitus dan hipertensi sekarang adalah penyebab utama penyakit ini (Faradilla, 2009). Kriteria yang terdapat pada penyakit ginjal kronik ini adalah timbulnya kerusakan ginjal lebih dari 3 bulan dengan kata lain terjadinya kelainan *structural* maupun *fungsional* (Faradilla, 2009). Adapun tanda dan gejala penyakit *Chronic Kidney Disease* (CKD) antara lain terjadinya kelainan pada urin terdapat dalam protein, sel darah putih/*leukosit*, darah/*eritrosit*, bakteri, *creatinine* darah naik, *hemoglobin* turun, *protein* yang selalu positif (Wariantio, 2011). Penyakit CKD ini dapat menyerang siapa saja yang menderita penyakit serius atau terluka dimana hal itu dapat berdampak langsung pada ginjal itu sendiri. Penyakit CKD lebih sering dialami mereka yang berusia dewasa, terlebih pada kaum lanjut usia.

2.3 Jaringan Saraf Tiruan (JST)

Komputer digital mengungguli di dalam perhitungan numeric dan manipulasi simbol yang terkait dibandingkan manusia. Jaringan saraf tiruan merupakan sistem komputasi parallel yang besar dan yang terdiri dari beberapa prosesor sederhana dengan banyak interkoneksi yang mencoba model untuk menggunakan beberapa prinsip “organisasi” yang diyakini digunakan oleh manusia (Jain dan Mao, 1996). Penelitian jaringan saraf tiruan telah mengalami tiga periode dalam

eksistensi aktifitas. Pada sekitar tahun 1940 oleh McCulloch dan Pitts menjadi yang pertama dalam penelitian ini, kedua terjadi pada sekitar tahun 1960 dengan teori konvergensi *perceptron* karya Minsky dan Papert's dapat mengurangi keterbatasan dari *simple perceptron*. Penelitian yang dilakukan ini membuat dampak berkurangnya antusias orang di dalam ilmu komputer dan membuat jeda penelitian tentang jaringan saraf tiruan berlangsung hampir 20 tahun sejak sekitar tahun 1980. Jaringan saraf tiruan mendapatkan banyak minat baru. Perkembangan dibalik kebangkitan ini adalah Hopfield pada tahun 1982 tentang pembelajaran algoritme propagasi balik untuk *perceptrons multilayer (multilayer feedforward network)* yang diajukan pertama kali oleh Werbos dan kemudian di populerkan oleh Rumelhart pada tahun 1986, Anderson dan Rosenfeld memberikan sejarah yang rinci pada perkembangan jaringan saraf tiruan (Jain dan Mao, 1996).

Funahashi dan Hornik beserta para peneliti lainnya telah menunjukkan bahwa jaringan saraf dengan kompleksitas yang memadai dapat mendekati fungsi apapun serta fungsi tidak dapat diketahui sampai tingkat akurasi yang diinginkan dengan hanya satu lapisan tersembunyi oleh karena itu, model JST dapat di representasikan yang terdiri dari lapisan input, lapisan tersembunyi dan juga lapisan *output* yang masing masing terhubung 1 sama lainnya (Mingyue, Cheng, dan Yu, 2016).

2.3.1 Fungsi Aktivasi Jaringan Saraf Tiruan

Dalam jaringan saraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu *neuron*, argument fungsi aktivasi adalah net input (kombinasi linear input dan bobotnya) (Siang, 2009). Beberapa fungsi aktivasi yang dipakai dalam jaringan saraf tiruan salah satunya fungsi sigmoid biner. Fungsi ini digunakan untuk jaringan saraf tiruan yang dilatih dengan menggunakan metode *backpropagation*. Fungsi *sigmoid biner* memiliki nilai pada *interval* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf tiruan yang membutuhkan nilai *output* yang terletak pada *interval* 0 sampai 1. Namun, fungsi ini juga digunakan oleh jaringan saraf tiruan yang nilai outpunya 0 atau 1.

Fungsi aktivasi yang sering digunakan dalam jaringan saraf tiruan diantaranya:

a. Fungsi *Sigmoid Biner*

Digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode *backpropagation*. Memiliki nilai pada *interval* 0 sampai 1.

Fungsi *sigmoid biner* dirumuskan:

$$y = f(x) = \frac{1}{1+e^{-ax}} \quad (2.1)$$

b. Fungsi *Sigmoid Bipolar*

Output dari fungsi ini memiliki *interval* antara 1 sampai -1 dan fungsinya dirumuskan:

$$y = f(x) = \frac{1-e^{-x}}{(1+e^{-x})} \quad (2.2)$$

2.4 Extreme Learning Machine (ELM)

Extreme Learning Machine adalah salah satu metode jaringan saraf tiruan feedforward dengan satu *hidden layer* yang biasanya sering disebut dengan istilah *single hidden layer feedforward neural network* (SLFNs) (Huang, Zhu, dan Siew, 2006). Metode ELM dapat mengatasi hal seperti kelemahan dari jaringan saraf *feedforward* terutama dalam hal learning speed. Algoritme ELM tidak melatih bobot input melainkan melatih untuk memperoleh bobot keluarannya menggunakan moore-penrose invers pada sistem linear secara umum. Dengan menemukan node yang memberikan nilai *output* maksimal dan parameter parameter seperti input weight dan bias dipilih secara *random*. Sehingga algoritme ELM dapat dipercaya memiliki learning speed yang optimal dan cepat yang mampu menghasilkan good generalization performance.

2.4.1 Algoritme ELM

Huang, Zhu, dan Siew (2006) memaparkan bahwa ELM memiliki tiga tahapan, yaitu jika diberi data *training*, fungsi aktivasi $g(x)$ dan m unit *hidden*, maka langkah-langkah training pada metode ELM yang akan diproses adalah sebagai berikut:

Langkah 1 : Inisialisasi semua bobot dan jumlah di dengan bilangan acak kecil $[-0,5,0,5]$. Dengan ukuran bobot W adalah $[\text{Hidden Neuron} \times \text{Input Layer}]$.

Langkah 2 : Hitung matrix inisialisasi *output hidden layer*

$$H_{init} = \begin{bmatrix} (x_{11} \cdot w_{11}^T) + bias & (x_{12} \cdot w_{12}^T) + bias & (x_{13} \cdot w_{13}^T) + bias \\ (x_{21} \cdot w_{21}^T) + bias & (x_{22} \cdot w_{22}^T) + bias & (x_{23} \cdot w_{23}^T) + bias \\ (x_{31} \cdot w_{31}^T) + bias & (x_{23} \cdot w_{23}^T) + bias & (x_{33} \cdot w_{33}^T) + bias \end{bmatrix}_{j \times k} \quad (2.3)$$

Keterangan:

x_{ij} = data pada baris ke i kolom ke j

w_{ij}^T = nilai bobot transpose pada baris ke i kolom ke j

Bias = nilai bias dengan nilai range $[0,1]$

Langkah 3 : Setelah menghitung nilai H_{init} mencari nilai matrix H

$$H = 1/(1 + EXP(-H_{init})) \quad (2.4)$$

Langkah 4 : Menghitung matrix H^+ setelah mencari nilai matrix H

Setelah mendapatkan matriks H dengan ukuran $n \times m$, selanjutnya hitung H^+ yang merupakan matriks *pseudoinverse* dari matriks H yang akan digunakan pada pencarian nilai bobot antara hidden layer dan output layer, persamaan H^+ sebagai berikut:

$$H^+ = (H^T H)^{-1} H^T \quad (2.5)$$

Kemudian mencari bobot ke *output layer* (β)

$$\beta = H^T t_i \quad (2.6)$$

Keterangan :

H^T = nilai H Transpose.

t_i = target dari proses training.

Langkah 5 : Menghitung nilai Y prediksi yang akan di bandingkan dengan nilai target.

$$\hat{Y} = H \cdot \beta \quad (2.7)$$

Langkah 6 : Mencari nilai max dari nilai Y prediksi yang di dapatkan, serta membandingkan nilai max dengan dengan nilai Y prediksi untuk dapat mengetahui hasil kelas nya.

Langkah 7 : Menghitung semua perubahan bobot dan jumlah hidden neuron yang optimal.

Langkah 8 : Simpan nilai β dari bobot dan jumlah *hidden neuron* yang optimal.

Setelah didapatkan proses *training* dilakukan proses *testing* yang dimana dengan tahapan sebagai berikut:

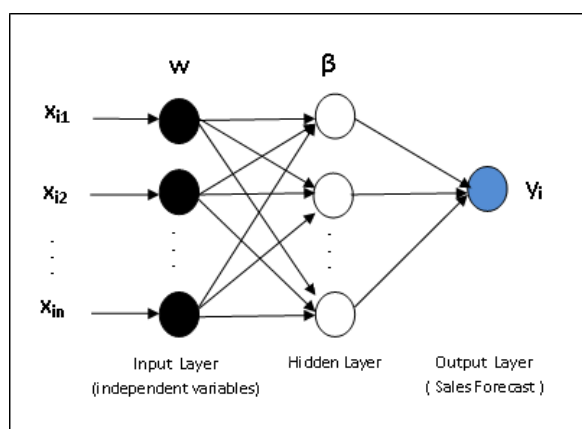
Langkah 1 : Mengambil nilai bobot, bias, *hidden neuron*, dan fungsi aktivasi yang digunakan dalam proses *training*.

Langkah 2 : Menghitung nilai keluaran *hidden neuron* dengan persamaan 2.3.

Langkah 3 : Menghitung nilai keluaran *hidden neuron* dengan fungsi aktivasi yang digunakan pada proses *training*.

Langkah 4 : Menghitung nilai Y prediksi yang akan dibandingkan dengan nilai target dengan menggunakan nilai bobot keluaran yang digunakan pada proses *training* dengan menggunakan persamaan 2.7.

2.4.2 Arsitektur *Extreme Learning Machine* (ELM)



Gambar 2.1 Contoh Arsitektur *Extreme Learning Machine* (ELM)

Sumber: (Qoid Humaini, 2015)

2.5 Confusion Matrix

Confusion Matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Kohavi dan Provost, 1998). Kinerja sistem seperti itu biasanya di evaluasi menggunakan data dalam matriks. Tabel berikut menunjukkan kinerja confusion matrix dengan 2 kelas klasifikasi.

		Ground Truth	
		CKD	Not CKD
Predicted	CKD	A	B
	Not CKD	C	D

Tabel 2.1 Contoh Penerapan *Confusion Matrix*

Input dari *confusion matrix* memiliki keterangan sebagai berikut:

- Jika hasil CKD dan data sebenarnya CKD
- Jika hasil prediksi Not CKD sedangkan nilai sebenarnya CKD
- Jika hasil prediksi CKD sedangkan nilai sebenarnya Not CKD
- Jika hasil prediksi Not CKD dan nilai sebenarnya CKD

Confusion matrix melakukan perhitungan dengan 4 keluaran yaitu *recall*, *precision*, *accuracy* dan *f1*.

Recall adalah tingkat keberhasilan suatu sistem dalam menemukan kembali sebuah informasi. *Recall* dapat dinyatakan dengan rumus:

$$\frac{A}{(A+C)} \quad (2.8)$$

Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Precision* dapat dinyatakan dengan rumus:

$$\frac{A}{(A+D)} \quad (2.9)$$

Akurasi adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual. Akurasi dapat dinyatakan dengan rumus:

$$\frac{(A+B)}{(A+B+C+D)} \quad (2.10)$$

F-Measure adalah perhitungan evaluasi yang mengkombinasikan antara nilai *precision* dan *recall*

$$\frac{(B+C)}{(A+B+C+D)} \quad (2.11)$$

2.6 Normalisasi Data

Dalam Melakukan proses perhitungan dalam klasifikasi penyakit *chronic kidney disease* (CKD), perlu dilakukan proses normalisasi data. Hal ini bertujuan untuk standarisasi semua data yang digunakan dalam perhitungan sehingga data berada pada jarak tertentu (Patro dan Sahu, 2015). Proses normalisasi yang digunakan pada penelitian ini adalah mentransformasikan pada *range* yang lebih kecil [0,1] dengan menggunakan metode *min max normalization* yang memiliki persamaan:

$$x' = \frac{(x-a)}{((b-a)+(1-0))} \quad (2.12)$$

Keterangan:

x' = Data yang sudah di transformasi

x = Data Awal

a = Data minimum dari keseluruhan data

b = Data maximum dari keseluruhan data